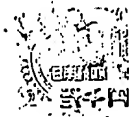


策抄記号 2-7



特許庁長官 齋藤英雄殿
イギリス国特許出願口
西暦 1973 年 7 月 18 日
34215/73

特 許 願

(200011)

昭和 49 年 7 月 17 日

特許庁長官 齋藤英雄殿

発明の名称

データ処理装置

発明者

住 所 イギリス国ランカシャー、ハスリングデン、
サンダウン、ロード、14

氏 名 ジョン、リチャード、イートン

特許出願人

(ほか 1 名)

住 所 イギリス国ロンドン、サウス、ウエスト、15、
ブットニー、アイ、シー、エル、ハウス、(番地なし)

名 称 インターナショナル、コンピュータース、リミテッド
代表者 ジョン、ロバート、カートライト

国 籍 イギリス国

代 理 人 (郵便番号 100)

東京都千代田区丸の内三丁目2番3号
(電話東京(211)232123)

4230 弁 理 士 猪 股

(ほか 2 名)

49-082104

万 式 登 記

明 細 書

発明の名称 データ処理装置

特許請求の範囲

記憶装置内の少くとも 2 つのスタック内に少くとも 2 種類の情報が、1 つのスタックに 1 種類の情報が組合されるようにして、書込まれ、これら少くとも 2 つのスタックはそれらに情報が加えられるにつれて別々のベースアドレスから互いに接近することを特徴とするデータ処理装置。

発明の詳細な説明

本発明はデータ処理装置に関し、かつ必ずしもそのみに限定されるものではないが、特に記憶装置内にプログラム材料のブロックをオーバーレイするための装置に関する。

データ処理装置に生ずる 1 つの問題は、いくつかの種類の情報に記憶スペースを割当てることである。この場合には各種類における割当てべき情報量がデータ処理装置の動作中に変化する。こ

(/)

① 日本国特許庁

公開特許公報

① 特開昭 50 - 43844

④ 公開日 昭 50. (1975) 4. 19

② 特願昭 49 - 82104

② 出願日 昭 49. (1974) 7. 17

審査請求 未請求 (全 8 頁)

庁内整理番号

5825 56

⑤ 日本分類

9700C0

⑤ Int. Cl²

G06F 13/00

のような場合に記憶スペースを割当て 1 つの方法は、情報の各種類に対して別々の固定記憶領域を与えることである。しかし、これは関連する情報の種類の全ての蓄積要求を満足できなければならないから、各記憶領域は比較的大きくなければならない。そうすると、ある与えられた時刻にはある種類の情報のみがそのような大位の記憶を行うことを要するのに対して、他の情報は非常に少いか、または全くないと予想されるから、記憶スペースにかなりの無駄が生ずることになる。この記憶スペースの無駄は、利用可能であればどの記憶スペースにでも書込めるように情報を構成することにより減少できる。そのためには、情報の各項目がどこに貯えられているかの記録を保持するための表と、記憶装置の使用を制御する比較的複雑な記憶装置管理装置とを備える必要がある。しかし、このようにすると同じ種類の情報が連続する場所に貯えられないで、記憶装置全体に分散されることになるが、このことはある状況の下では、すなわち通常は順次実行されるマイクロプログラムが

(2)

情報の場合に不利となる。

本発明の目的は、データ処理装置において蓄積スペースを割当てる新規な技術を提供することである。

本発明によれば、記憶装置内の少くとも2つのスタック内に少くとも2種類の情報が、1つのスタックに1種類の情報が組合されるようにして、書込まれ、これら少くとも2つのスタックはそれらに情報が加えられるにつれて別々のベースアドレスから互いに接近するようにされるデータ処理装置が得られる。

このような装置では、2種類の情報は同じ記憶スペースを共有するが、2種類の情報の総計要求が利用可能なスペースよりも大きい時だけ、それら2種類の情報の蓄積要求が衝突する。このように記憶スペースを共有することによつて、別々の記憶領域が備えられている場合に必要とする総計記憶スペースよりも記憶スペースを小さくできる。しかし、各種類の情報は別々のスタックを有するから、そのスタック内の情報はなお連続する場所

(3)

記憶装置から読出されるマイクロプログラムのブロックをそなえているような装置に特に応用できる。

以下、図面を参照して本発明を詳細に説明する。

まず第1図を参照して、図示の装置はデータとマイクロプログラム材料を保持するための主記憶装置10と、マイクロプログラム記憶装置11と、マイクロプログラム制御ユニット12とを有する。動作の時には制御ユニット12は主記憶装置10からプログラム命令の取り出しを行い、各命令に対して、その命令を実行するためにマイクロプログラム記憶装置11からのマイクロプログラム命令の適切な順序づけを開始する。データ処理装置のそのようなマイクロプログラム制御はもちろん周知であり、またいずれにしてもマイクロプログラム制御ユニット12は本発明の構成部分ではない。

マイクロプログラム記憶装置11の容量は主記憶装置10の容量よりも小さいが、マイクロプログラムユニットのためのマイクロ命令を迅速にアクセスできるように、アクセス時間は主記憶装置より

(5)

に保持できる。

複数のスタックがかち合った時に、その中の1つのスタックが他のスタックに重ねて書くことができるように、その1つのスタックがなるべく優先権を持つようにする。本発明の一実施例では、優先度の高いか、もしくは低いスタックのいずれにも情報を加えるための利用可能なスペースが見出されないとする、優先度の低い全てのスタックを除去して、加えるべき情報のためのスペースを与える。

本発明の別の実施例では、第3のベースアドレスからスタートする第3のスタックに書込まれ、情報が加えられるにつれて他の2つのスタックへ向つて進む、少くとも第3の種類の情報スタックを設けることができる。この第3のスタックは初めの2つのスタックのうちの少くとも1つのスタックよりも高い優先度を持つようにすると便利である。

本発明は、記憶装置がマイクロプログラム記憶装置であり、かつその記憶装置に書込む情報が主

(4)

もはるかに短い。マイクロプログラム記憶装置11の1つの領域13は、データ処理装置の基本制御のために要求される基本マイクロプログラム材料（「プリミティブインタフェース」と呼ばれる）のために保留される。このマイクロプログラム材料はマイクロプログラム記憶装置内に永久的に保持される。マイクロプログラム記憶装置の残りの領域14は、データ処理装置により現在使用されている付加的なマイクロプログラム材料のいくつかのブロックのコピーの保持のために利用できる。主記憶装置10の1つの領域は、データ処理装置のマイクロプログラムの全てのブロックのマスタコピーを保持するためのバックアップ記憶装置として機能する。マイクロプログラムユニット12で使用するために、求められた時にはそれらのブロックの任意の1つをマイクロプログラム記憶装置へ転送できる。転送されたブロックはそのマイクロプログラム記憶装置に既に貯えられているある情報にオーバレイされる。この理由からマイクロプログラムのブロックを以後「オーバレイ」と呼ぶ

(6)

ことにする。第1図で、そのような1つのオーバーレイの主記憶装置10におけるマスタコピーは斜線のつけた領域15で示し、マイクロプログラム記憶装置における対応するコピーは斜線をつけた領域16により示す。

オーバーレイのためにこのバックアップ領域と、マイクロプログラム記憶装置をオーバーレイするための装置とを設けることにより、このデータ処理装置は極めて高価な高速大容量のマイクロプログラム記憶装置を備える必要なしに、利用可能な大量のマイクロプログラムを持つことができる。

ここで説明している実施例では、マイクロプログラムオーバーレイは次のような2つの種類に分類される。

- (i) システム・オーバーレイ。これらは、データ処理装置の範囲を拡張し、効率を高めるためにプリミティブ・インタフェース材料の拡張を実際に構成するマイクロプログラム材料のブロックである。たとえば、それらのシステム・オーバーレイはページめくりのような管理機能を実行で

(7)

長レジスタ19がそれである。オーバーレイ表17はシステム内の各オーバーレイのために1つのエントリを含む。各エントリは次のものを有する。

- (i) オーレイの長さ(すなわち、オーバーレイにおけるマイクロ命令の数)を定めるフィールドVL。一般に、異つたオーバーレイは異つた長さを有する。
- (ii) マイクロプログラム記憶装置におけるオーバーレイのスタートアドレスを定めるフィールドVA。オーバーレイがマイクロプログラム記憶装置内に存在しないとすると、このフィールドは零にセットする。
- (iii) 主記憶装置内におけるオーバーレイのマスタコピーのスタートアドレスを定めるフィールドVSA。

第1図にはオーバーレイコピー15、16に対するそのような1つの表エントリ20を示す。この表エントリではフィールドVL、VA、VSAとオーバーレイ15、16との間の関係を矢印で示してある。

このデータ処理装置のプログラムがある特定の

(9)

き、あるいはエミュレーション、すなわち、指令コードとシステム構成の異なる他の機械を真似るために必要とされる。

- (ii) ユーザー・オーバーレイ。これらは特殊な用途たとえば平方根ルーチンにおいてしばしば要求されることがある、特殊なタスクを実行するためのマイクロプログラム材料のブロックである。一般に、これらのオーバーレイはシステムの製作者よりもユーザーにより書き込まれる。

この分類はある範囲までは明らかに任意であり、かつ全く便宜上行われたものであると考えるべきである。

主記憶装置10とマイクロプログラム記憶装置11との間のオーバーレイの転送は、オーバーレイ表17の使用により制御される。このオーバーレイ表は実際には主記憶装置10の一部であつて、次の2つのレジスタにより定められる。すなわち、主記憶装置10内のオーバーレイ表のスタートのアドレスVTBAを含むオーバーレイ表ベースアドレス・レジスタ18と、オーバーレイ表の長さVTLを含むオーバーレイ表

(8)

マイクロプログラム・オーバーレイを使用することを要求する場合には、装置は記述子を記述子レジスタ21に置くことを含む呼出し命令を出す。この記述子は次のものを有する。

- (i) オーレイの形式を定める単一ビットVT。
VT=0 はユーザーオーバーレイを示し、VT=1 はシステムのオーバーレイを示す。
- (ii) 要求されたオーバーレイに関連する、エントリイのオーバーレイ表における位置を識別するフィールドVN。

フィールドVN は比較器22に加えられる。この比較器はそのVN の値をレジスタ19からのオーバーレイ表長VTLと比較する。VN がVTLよりも大きいとすると誤差が生じており、したがってプリミティブ・インタフェース13に適切な割り込みルーチン内へのエントリイを起させるように、割り込み信号が出力ライン23に与えられる。しかし、VN がVTLよりも大きくないと仮定すると、VN の値は加算器24に加えられ、そこでレジスタ18からの値VTBAに加え合わされて、オーバーレイ表17にお

(10)

ける適切なエントリイのアドレスを形成する。エントリイのフィールドVAが読出され、マイクロプログラム記憶装置11のアドレスに使用される。要求されたオーバレイのコピーが実際はマイクロプログラム記憶装置11に現在あるとすると、そのためにその記憶装置内でオーバレイのスタートへの飛越しが行われる。しかし、要求されたオーバレイのコピーが現在マイクロプログラム記憶装置内にないとすると、VAの値は0となるであろうからマイクロプログラム記憶装置はその零アドレス場所でアクセスされる。この場所はプリミティブ・インタフェース13内で特殊なオーバレイ・ルーチンへの飛越しを起させる飛越し命令を含む。それは主記憶装置10からの要求されたオーバレイのコピーのマイクロプログラム記憶装置11へのローディングを制御する。

次に第2図を参照して、オーバレイ・ルーチンは主記憶装置10からのオーバレイを、その種類に従ってマイクロプログラム記憶装置11内の2つのスタック25, 26の中に置く。システム・オーバレイ (11)

トにおける第1のフリーアドレスへのポインタ。

SP - システムオーバレイ・スタック25のフロントにおける第1のフリーアドレスへのポインタ。

SB - システムオーバレイ・スタックのベースアドレス。

ST - システムオーバレイ・スタックにおけるシステムオーバレイの総数。

これらのレジスタとマイクロプログラム内の場所との関係を第2図に矢印で示す。

UP レジスタと SP レジスタの内容は減算回路28で減算されて1だけ増分され、値 $X = UP - SP + 1$ を発生する。これは別のオーバレイを2つのスタック25, 26のフロントの中と間に書き込むために利用可能な自由スペースを表す。

オーバレイルーチンの最初のアクションは、記述子の種類を決定するために、記述子レジスタ21 (第1図) 内のVTフィールドの内容を検査する。もし $VT = 0$ とすると (ユーザーオーバレイを示

(13)

いはスタック25の中に置かれる。スタック25はベースアドレスSBからマイクロプログラム記憶装置内で上方向 (すなわち、アドレス値が増加する向き) へ伸びる。ユーザーオーバレイはスタック26の中に置かれる。スタック26はベースアドレスUBからマイクロプログラム内で下方へ伸びる。ベースアドレスUBは記憶装置11の上限である。したがって、これらのオーバレイが2つのスタックに加えられにつれて、それらのオーバレイは最終的に出会うまで互いに接近する。2つのオーバレイが会すると、システムオーバレイ・スタック25が優先し、後述するようにユーザーオーバレイ・スタック26を書き直すことができる。

オーバレイルーチンは1組のレジスタ27を使用する。これらのレジスタは実際にはオーバレイ17の第1場所に位置させることができる。これらのレジスタはそれぞれ次のような値を含む。

UB - ユーザーオーバレイ・スタック26のベースアドレス。

UP - ユーザーオーバレイ・スタックのフロント (12)

す)、第3図に示すオーバレイルーチンの部分が実行され、 $VT = 1$ とすると (システムオーバレイを示す)、第4図に示すオーバレイルーチンの部分が実行される。

次に第3図を参照して、ユーザーオーバレイの場合にはオーバレイ表17内の現在アドレスされているエントリイからのVTの値が、回路28からの値Xと比較され (ブロック30) て、新しいオーバレイを保持するためにスタックフロントの間で、マイクロプログラム記憶装置内に利用可能な十分な自由スペースがあるかどうかを決定する。VLがXよりも小さいか、または等しいとすると、オーバレイはユーザーオーバレイ・スタックを下方に伸ばすように、マイクロプログラム記憶装置のUPまで場所 $UP - VL + 1$ 内に直ちにロードできる (ブロック31)。それと同時に、オーバレイ表17は新しいオーバレイのスタートアドレス $UP - VL + 1$ をフィールドVAの中に書き込むことにより更新される。最後に、ポインタ・アドレスレジスタUPはそれから値VLを引くことにより更

(14)

新される(ブロック32)。これによりこの場合のオーバーレイルーチンは完了される。

次に、VL がXよりも大きいことが見出されたとすると、明らかに新しいオーバーレイは利用可能なスペース内に当てはまらない。この新しいオーバーレイのための余地を作るために、現在ユーザーオーバーレイ・スタック26内にある全てのオーバーレイが除去される(ブロック33)。各オーバーレイが除去されると、オーバーレイ表17におけるその対応するエントリがフィールドVA を零にセットすることにより更新され、オーバーレイがもはやマイクロプログラム記憶装置の中に入らないことを示す。それからポインタUP をUB に等しくセットすることによりポインタUP は更新される(ブロック34)。VL の値がXと再び比較される(ブロック35)。全てのユーザーオーバーレイの除去後でもVL が大きすぎるとすると、オーバーレイルーチンによりそれ以上何も行うことはできず、割込み信号が発生される。しかし、今ではVL がXよりも小さいか、または等しいとすると、オーバーレイ

(15)

いとすると、新しいシステムオーバーレイは利用可能なスペース内に明らかに入らない。しかし、システムオーバーレイ・スタックはユーザーオーバーレイ・スタックよりも優先権を持つているから、新しいシステムオーバーレイのための余地を作るために、ユーザーオーバーレイ・スタック26内に現在入っている全てのオーバーレイが除去される(ブロック43)。各オーバーレイが除去されると、フィールドVA を零にセットすることによつてオーバーレイ表17におけるその対応するエントリが更新される。それからポインタUP をUB に等しくセットすることによりポインタUP は更新される(ブロック44)。VL の値はXと再び比較される(ブロック45)。全てのユーザーオーバーレイが除去されてもVL がなお大きすぎるとすると、割込み信号が発生される。しかし、VL がXよりも小さいか、または等しいとすると、オーバーレイルーチンは前と同様に完了できる(ブロック41、42)。

以上の説明から、新しいユーザーオーバーレイまたはシステムオーバーレイのいずれかにより、ユー

(17)

ーチンは前記したように完了できる(ブロック31、32)。

次に第4図を参照する。システムオーバーレイの場合には、オーバーレイのための十分な自由スペースがあるかどうかを決定するために、VL の値がXと再び比較される(ブロック40)。VL がXよりも小さいか、または等しいとすると、システムオーバーレイ・スタックを上方へ延ばすように、オーバーレイをマイクロプログラム記憶装置の場所SP からSP + VL - 1まで直ちにロードできる(ブロック41)。それと同時に、新しいオーバーレイのスタートアドレスSP をフィールドVA の中に書き込むことにより、オーバーレイ表17は更新される。最後に、ポインタ・アドレスレジスタSP は値VL をそれに加えることにより更新され(ブロック42)、値ST (スタック内におけるシステムオーバーレイの数)は1だけ増分される。これによつてこの場合に対するオーバーレイルーチンは完了される。

次にブロック40に戻つて、VL がXよりも大き

(16)

ザーオーバーレイにより占められているスペースが要求される場合に、ユーザーオーバーレイはオーバーレイルーチンにより自動的に除去される。一方、システムオーバーレイはマイクロプログラムのプリミティブ・インタフェース中の対応するルーチンを開始させる特殊な「クリア・システムオーバーレイ」命令により除去できるだけである。このようにして、「最後に入れ、最初に出す(last in, first out)」を基準にして、システムオーバーレイの希望する任意の数を除去できる。除去すべき数Rは命令により指定される。

次に、「クリア・システムオーバーレイ」命令を実行するためのマイクロプログラムルーチンを示す第5図を参照する。最初のステップはRの値(除去すべきシステムオーバーレイの数)とST(マイクロプログラム記憶装置内のシステムオーバーレイの数)とを比較することである(ブロック57)。RがST よりも大きいとすると、明らかに誤りが生じており、適切な割込みが発生される。もしそうでないと、次のステップはRが零に等し

(18)

いかどうかを調べることである(ブロック52)。
Rが零でないとする、次のステップは、システムオーバーレイ・スタック25のフロントから1つのシステムオーバーレイを除去し、フィールドVAを零にセットすることにより対応するオーバーレイ表エントリを更新することである(ブロック53)。
次に、除去されたオーバーレイの長さVLをSPから引き、STを1だけ減分させることによりレジスタ27が更新される(ブロック54)。Rの値も1だけ減分される。それからRが今では零かどうかを調べるためにブロック52への復帰が行われる。もしRが零に等しいとすると、システムオーバーレイの要求された数が今では除去されており、したがってルーチンは完了されている。もしRが零に等しくないとする、Rが零になるまでループ53、54、52のステップが反復される。

1つまたはそれ以上のシステムオーバーレイをプリミティブ・インタフェースの一部として一時的に処理させるために(すなわち、スタックからこれらのオーバーレイが除去されることを防ぐ)、適
(19)

より下のプリミティブ・インタフェース材料(またはそのようなものとして一時的に取扱われるシステムオーバーレイ材料)を書き直すことは許されない。

スタック61のベースアドレスEBと、スタック61のフロントで第1の自由場所を指すポインタアドレスEPとを保持するために、セット27内には付加的な3つのレジスタが設けられる。レジスタ21(第1図)における記述子は、3種類のオーバーレイ形式を識別するために、今は2ビットフィールドVTを持たねばならず、オーバーレイルーチンはエミュレーション・オーバーレイのローディングを取扱うために拡張せねばならない。更に、第5図に示すのに類似する「クリヤ」ルーチンをエミュレーションオーバーレイのクリヤのために設けることができる。

以上説明した装置の別の改変では、この装置は2つの別々の処理ユニットを有する。これらのユニットは同じマイクロプログラム記憶装置11を共有し、各ユニットにはそのマイクロプログラムを

(20)

切な命令に応じてベースアドレスSBを変更するための装置を設けることができる。ベースアドレスSBがこのようにして変更される時にSTの値も変更せねばならない。

次に第6図を参照して、以上説明した装置の別の例においては、オーバーレイの第3の種類が要求に応ずることができる。この第3の種類はたとえば、システムオーバーレイの一部として先に考えたエミュレーション・オーバーレイを構成できる。この例ではエミュレーション・オーバーレイはマイクロプログラム記憶装置内の第3のスタック内に格込まれる。第3のスタック61はユーザーオーバーレイ・スタックのベースアドレスUBよりも高いベースアドレスEBからスタートし、他の2つのスタックへ向つて下向きに進む。なるべくなら、エミュレーションオーバーレイ・スタック61はユーザーオーバーレイ・スタック25と、システムオーバーレイ・スタック26よりも高い優先度を持ち、それによりそれらのスタックを書直すことができる。しかし、エミュレーションオーバーレイはアドレス

(21)

含むためにマイクロプログラム記憶装置の別々の領域が割当てられる。これらのユニットは主記憶装置10も共有する。この場合には、オーバーレイ表17は拡張されるから各エントリは今は処理ユニットの1つに関連するオーバーレイに対する1組のフィールドVL、VA、VSAと、他のユニットに関連するオーバーレイに対する同様な1組のフィールドとを含む。更に、各処理ユニットに1つずつ、合計して2組のレジスタ27を設けねばならない。

以上本発明をマイクロプログラム記憶装置におけるマイクロプログラムをオーバーレイすることに関連して説明してきたが、2種類またはそれ以上の情報が記憶装置に格込まれるような多くの場合により一般的に応用できることがわかるであろう。

図面の簡単な説明

第1図は本発明の装置の一部を示す概略ブロック図、第2図は本発明の装置の別の部分の概略ブロック図、第3〜5図は本発明の装置のマイクロプログラムを示し、第6図は本発明の装置の別の

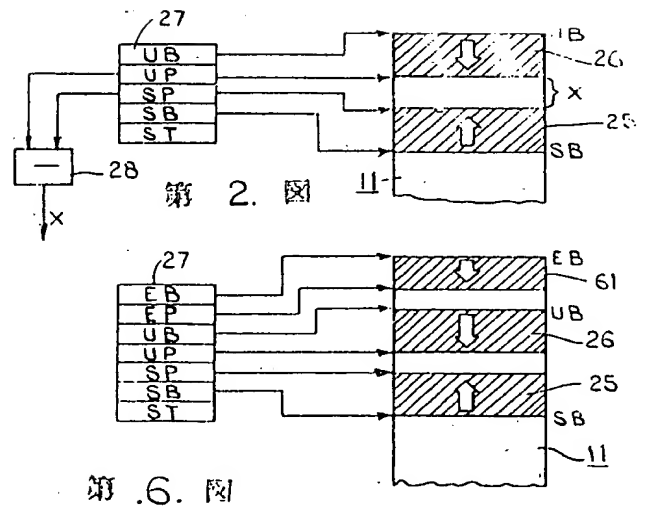
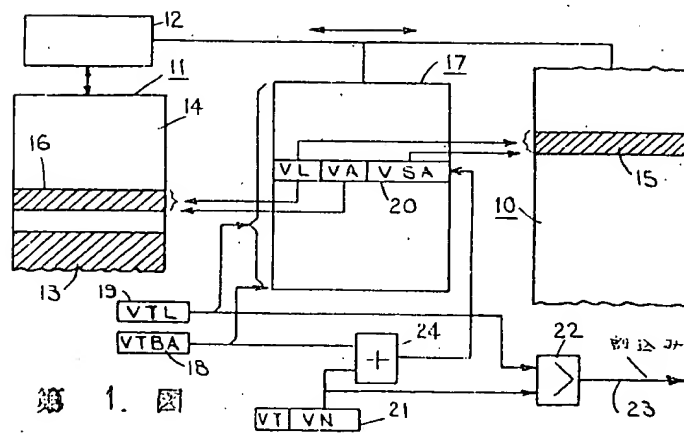
(22)

実施例の概略ブロック図である。

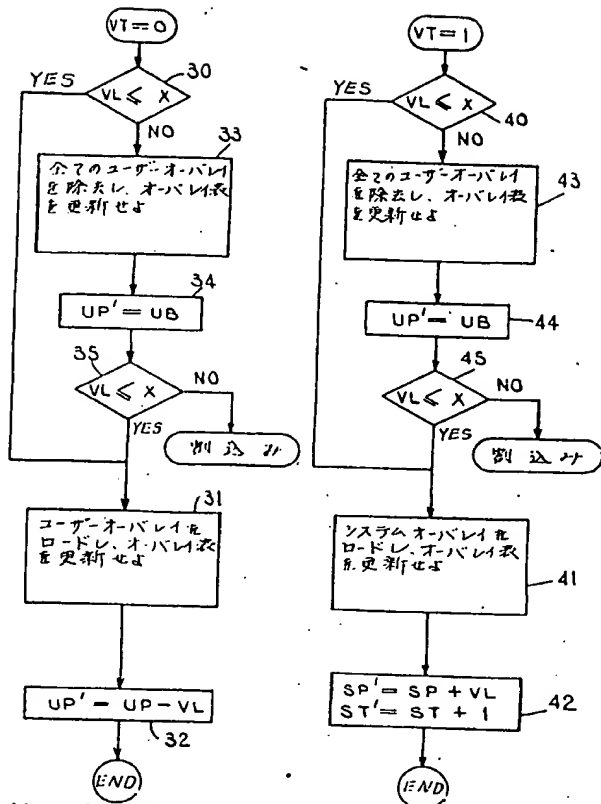
10…主記憶装置、11…マイクロプログラム記憶装置、12…マイクロプログラム制御ユニット、13、14、15、16、61…記憶領域、17…オーバーレイ表、18…オーバーレイ表ベースアドレスレジスタ、19…オーバーレイ表長レジスタ、20…表エントリ、21…記述子レジスタ、22…比較器、24…加算器、25、26…オーバーレイ・スタック、27…レジスタ。

出願人代理人 猪 股 清

(23)

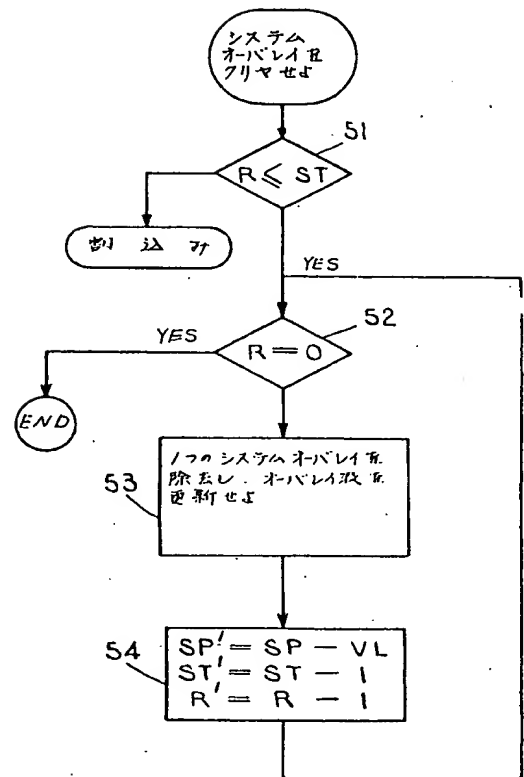


第 6. 図



第 3. 図

第 4. 図



第 5. 図

添 附 書 類 の 目 録

- | | |
|-------------------|-------|
| (1) 明 細 書 | 1 通 |
| (2) 図 面 | 1 通 |
| (3) 委任状およびその訳文 | 各 1 通 |
| (4) 優先権証明書およびその訳文 | 各 1 通 |

前記以外の発明者、特許出願人または代理人

発 明 者

住 所 イギリス国ランカシャー、ハスリングデン、
サンダウン、ロード、23
氏 名 フィリップ、ロナルド、ブラディー

代 理 人 (郵便番号 100)
東京都千代田区丸の内三丁目2番3号

3202 弁 理 士 佐 藤 勇

同 所

6428 同 佐 藤 一

同 所

同



-226-